BUNDESREPUBLIK DEUTSCHLAND

JC997 U.S. PTO
09/902855
07/12/01

Bescheinigung

Das Forschungszentrum Karlsruhe GmbH in Karlsruhe, Baden/Deutschland hat eine Patentanmeldung unter der Bezeichnung

"Digital-elektronisches Verfahren zur Steigerung der Berechnungsgenauigkeit bei nichtlinearen Funktionen und eine Hardware-Architektur zur Durchführung des Verfahrens"

am 15. Februar 1999 beim Deutschen Patent- und Markenamt eingereicht.

Das angeheftete Stück ist eine richtige und genaue Wiedergabe der ursprünglichen Unterlage dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig das Symbol G 06 F 7/52 der Internationalen Patentklassifikation erhalten.

München, den 9. März 2000

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Nietiedt

Aktenzeichen: 199 06 559.4

Forschungszentrum
Karlsruhe GmbH
ANR 5661498

Karlsruhe, den 12. Feb. 1999
PLA 9906 Mh/he

Digital-elektronisches Verfahren zur Steigerung der Berechnungsgenauigkeit bei nichtlinearen Funktionen und eine Hardware-Architektur zur Durchführung des Verfahrens

Forschungszentrum
Karlsruhe GmbH
ANR 5661498

Karlsruhe, den 12. Feb. 1999
PLA 9906 Mh/he

Patentansprüche:

1. Digital-elektronisches Verfahren zur Steigerung der Berechnungsgenauigkeit bei nichtlinearen Funktionen, bestehend aus den Schritten:

   - der zur Bearbeitung als Zahl anstehende Wert aus einer im allgemeinen nichtlinearen Funktion wird als Eingabewort zusammen mit einem zugehörigen, kodierten Steuerwort $_f$ mit dem Eingabeformat
     $EF_f = S\ Ü1_f\ M_f\ A1_f$,
     und dem Komma an nicht festgelegter Stelle,
     wobei S die Vorzeichenstelle, $Ü1_f$ die Stellen mit dem höchsten Wert, die voraussichtlich niemals gesetzt werden können, M_f die Stellen mit der einheitlichen Breite m und $A1_f$ die Stellen mit dem niedrigsten Wert, die nicht verwendet werden können, und der Index „$_f$" das kodierte Steuerwort der Länge F ist,
     wird in eine erste Multiplexereinrichtung M1 einer elektronischen Datenverarbeitungseinrichtung mit $2^F = f$ Eingängen zu je m Stellen eingegeben und
   - in dieser auf ein Zwischenformat
     $ZF = S\ Ü2_c\ B_c\ A2_c$
     mit (m + 1)-Stellen und einer festen Kommastelle (Festkommadarstellung) abgebildet werden, wobei die Stellen $Ü2_c$ in der Überlaufeinrichtung Ü auf einen Überlauf hin überprüft und die niederwertigen Stellen $A2_c$ in der elektronischen Abschneideeinrichtung A abgetrennt werden,
   - der Zahlenbereich, der durch das Zwischenformat ZF am Ausgang der ersten Multiplexereinrichtung M1 dargestellt wird, wird in C teilweise unterschiedlich große Intervalle unterteilt, die den gesamten Zahlenbereich von ZF ohne Überlappung und ohne Lücken überdecken,

- das Zwischenformat ZF wird in einen Bereich K_c zur Kodierung und in einen Bereich G_c niederwertiger Stellen unterteilt, wobei beide Bereiche sich überlappen können,
- die Kodierung K_c wird in einer elektronischen Kodierungseinrichtung K aus B_c durchgeführt,
- an die vorzeichenbehaftete Kodierung SK_c werden die niederwertigen Stellen G_c angehängt,
- in einer weiteren Multiplexereinrichtung M2 mit C Eingängen der Breite des Ausgangsformats AF wird elektronisch die Ausgabeabbildung

  $K_cG_c \rightarrow KG$

  durchgeführt, so daß ein einheitliches Ausgabeformat

  AF = S KG

  vorliegt.

2. Hardware-Architektur zur Durchführung eines digital-elektronischen Verfahrens zur Steigerung der Berechnungsgenauigkeit bei nichtlinearen Funktionen, bestehend aus den folgenden Baugruppen:
   - einer ersten Multiplexereinrichtung M1 mit 2^F Eingängen zur Eingabe beliebiger, durchnummerierbarer Eingangsformate festgelegter Wortbreite m mit dem Festkomma an unterschiedlicher Stelle,
     mit einem weiteren kodierten Steuereingang, über den die durchnummerierten Eingabeformate EF_f angewählt werden können,
     mit einem Ausgang, an dem ein einheitliches Zwischenformat ZF ebenfalls festgelegter Wortbreite, das das Festkomma an nur noch einer vorgegeben, festgelegten Stelle aufweist;
   - einer Überlaufeinrichtung Ü, in die einerseits die höchstwertigen Stellen $Ü1_f$ des Eingabeformats EF_f, die voraussichtlich niemals gesetzt werden, und in die andrerseits die höherwertigen Stellen $Ü2_c$ des Zwischenworts ZF_m an der ersten Multiplexereinrichtung M1, die auf einen Überlauf hin überprüft werden müssen, eingegeben und

darin auf von Null verschiedene Stellen abgefragt werden, um dann im Falle gesetzter Stellen einen Alarm ausgeben zu können;

- einer Kodierungseinrichtung K, in dem aus dem zu kodierenden Teilbereich B_C des Zwischenformats ZF_m der kodierte Bereich K_C erzeugt wird;
- einer Abschneideeinrichtung A, in der die niedrigstwertigen Stellen $A1_f$ und die niederwertigen Stellen $A2_C$ von der weiteren Datenverarbeitung ausgeschlossen werden;
- einer zweiten Multiplexereinrichtung M2, in der der vorzeichenversehene, kodierte Bereich SK_C und angehängte, unkodierte Bereich G_C der niederwertigen Stellen im Zwischenformat ZF in ein vorgegebenes Ausgabeformat AF transformiert werden.

3. Digital-elektronisches System nach Anspruch 2, dadurch gekennzeichnet, daß die Überlaufeinrichtunt Ü, die Kodiereinrichtung K, die Abschneideeinrichtung A aus logischen Bausteinen aufgebaut sind.

4. Digital-elektronisches System nach Anspruch 3, dadurch gekennzeichnet, daß das System in einen spezifischen Chip oder in einen spezifischen Chipsatz gefaßt ist.

Beschreibung

Die Erfindung betrifft ein Verfahren zum Betreiben eines elektronischen Systems mit dem die Berechnungsgenauigkeit bei nichtlinearen Funktionen gesteigert wird.

Rechenintensive Algorithmen, wie sie insbesondere in der Bild- und Signalverarbeitung verwendet werden, werden im allgemeinen auf einem Computer oder, oftmals bei technischen Anwendungen, auf Mikroprozessoren oder Digitalen Signalprozessoren ausgeführt. Die Ausführungszeit auf diesen Prozessoren ist oft sehr lange, so daß zur Beschleunigung der Abarbeitung rechenintensiver Algorithmen der Einbau einer spezifischen Hardware sinnvoll ist. Das bedeutet, daß ein spezieller Chip (oft ein kundenspezifischer Chip - ASIC) oder Chip-Satz auf einer elektronischen Karte untergebracht wird, die beispielsweise in einem Personalcomputer zur Beschleunigung verwendet wird. Unterschiede im Vergleich zur herkömmlichen Prozessorlösung betreffen vor allem die Datenformate.

Als Zahlenformat weist eine Festkomma-Darstellung im Vergleich zu einer Gleitkomma-Darstellung den Vorteil einfacher und schneller Berechnung auf, weshalb in den meisten benutzerspezifischen Chips (ASIC) diese Darstellungsart verwendet wird. Der größte Nachteil dabei ist die reduzierte Genauigkeit im Vergleich zu Gleitkomma-Operationen. Bei der internen Berechnung wird daher bei der Zahlendarstellung oft auf eine größere Wortbreite übergegangen, die aber an der externen Schnittstelle nicht beibehalten werden kann, da dann der Aufwand für die Datenspeicherung zu groß wird (siehe The IEEE standard for binary floating point arithmetic, ANSI/IEEE Standard 754 - 1985). Dieses Fließkomma-Format ist sehr allgemein und daher hinsichtlich der Größe und Zahl der Bauelemente lange nicht so effizient einzubauen bzw. unterzubringen.

Wird für das Zwischenformat eine größere Wortbreite verwendet als für das Ausgabeformat, muß eine Umwandlung vom größeren zum

kleineren Format geschehen. Dies ist mit einer Genauigkeitseinbuße verbunden. Im allgemeinen werden hierfür so viele Bits des längeren Wortes abgeschnitten, bis der Inhalt in das kürzere Wort paßt. Wird bei den hochwertigen Bits gekürzt, muß für eine entsprechende Überlauf-Behandlung gesorgt werden, wird bei den niederwertigen Bits zu stark gekürzt, leidet die Genauigkeit darunter. Wird nicht weiter beachtet, wie die Daten extern des Chips weiterverarbeitet werden, gibt es im allgemeinen keine andere Methode, die Genauigkeit der Ausgabedaten zu erhöhen.

Der Erfindung liegt daher die Aufgabe zugrunde, ein Verfahren zum Betreiben eines elektronischen Systems bereit zu stellen, mit dem die Berechnungsgenauigkeit bei nichtlinearen Funktionen gesteigert wird, und das elektronische System zu realisieren, mit dem das Verfahren zeitoptimal durchgeführt werden kann.

Die Aufgabe wird durch die im Anspruch 1 aufgeführten Verfahrensschritte gelöst.

Zur allgemeinen Erläuterung wird kurz der Spezialfall umrissen, daß die Daten extern durch ein Modul, das eine nichtlineare Funktion realisiert, weiterverarbeitet werden. Es handelt sich beispielsweise um eine Lookup-Tabelle, die einen Eingabewert auf seinen Funktionswert abbildet. Dieser Fall kommt häufig vor, wenn komplizierte Funktionen sehr schnell berechnet werden sollen. Ein Beispiel ist die Berechnung eines Neuronalen Netzes, die im wesentlichen durch Matrix-Multiplikationen und eine anschließende nichtlineare Übertragungsfunktion erfolgt. Die Matrixmultiplikation kann effizient auf einem anwendungsspezifischen Mikrochip realisiert werden. Die nichtlineare Funktion, beispielsweise der Tangenshyperbolikus, wird durch eine Lookup-Tabelle dargestellt. Bei einer solchen Konstellation ist erst die Genauigkeit am Ausgang der Lookup-Tabelle entscheidend. Im Falle nichtlinearer Funktionen ist sie aber wesentlich geringer als die Genauigkeit der vom Chip stammenden, bereits reduzierten Daten.

Ein einfaches Beispiel demonstriert das:
Angenommen, die Tabelle stelle die quadratische Funktion $f(x) = x^2$ dar. Zum einfacheren Verständlichkeit wird statt einer binären eine dezimale Zahldarstellung herangezogen. Es interessiert der Bereich im Intervall [0,1]. Die vom Chip stammenden Daten haben eine Genauigkeit von 0.1, die aus der Tabelle resultierenden Daten ebenfalls. Die Tabelle bildet alle möglichen Zahlen des Formats auf f(x) ab, d.h. die Tabelle besitzt 11 Einträge. Die drei niedrigsten Werte {0, 0.1, 0.2} werden - exakt gerechnet - auf {0, 0.01, 0.04} abgebildet. Da die Zahlengenauigkeit aber nur 0.1 beträgt, werden alle drei Werte auf den neuen Wert 0 abgebildet. Auf der anderen Seite wird 0.9 auf 0.8 und 1 auf 1 abgebildet. Der Wert 0.9 kann im Bildbereich gar nicht vorkommen. Durch die Quantisierung wird hier also ein maximaler Fehler von 0.2 erzeugt, der einerseits durch die begrenzte Wortbreite der vom Chip stammenden Daten ausgelöst wird, andererseits durch die nichtlineare Funktion der Lookup-Tabelle vergrößert wird. Abhilfe schafft hier eine Kodierung der Daten, die als Eingabe für die Lookup-Tabelle günstiger ist.

Die Lösung liegt nun darin, daß das genaue Eingabeformat zwar eine festdefinierte Wortbreite besitzt, aber das Festkomma sich an unterschiedlichen Stellen befinden darf. Jede einzelne Position des Festkommas entspricht einem eigenen Format. In einem ersten Schritt wird aus diesen verschiedenen Formaten ein einheitliches Format hergestellt, welches das Festkomma an einer definierten Stelle aufweist. Da hierfür schon einige höherwertige Bits abgeschnitten werden können, kann ein Überlauf auftreten, der behandelt werden muß. Die Herstellung des einheitlichen Formats wird durch einen Multiplexer realisiert, der als Eingabe die unterschiedlichen Formate erhält und als Ausgabe das Einheitsformat ausgibt. Die unterschiedlichen Formate werden durchnummeriert und durch einen kodierten Steuereingang des Multiplexers angewählt.

Im zweiten Schritt wird der gesamte Definitionsbereich in mehrere Unterbereiche eingeteilt, die jeweils für sich eine sepa-

rate Zahlendarstellung benutzen. Fürs Weitere wird eine binäre Zahldarstellung zugrunde gelegt und als ganze Zahl interpretiert, so daß 1 der kleinste Unterschied zwischen zwei verschiedenen Zahlen ist. Dann läßt sich für die Unterteilung des Definitionsbereichs in Unterbereiche erreichen, daß für „flache" Funktionsbereiche, in denen die erste Ableitung viel kleiner als 1 ($f'(x) << 1$, für alle x aus dem betreffenden Unterbereich) eine weniger genaue Zahldarstellung gewählt wird. Der Grund dafür ist, daß der Bildbereich B des Funktionsabschnitts kleiner ist als der Definitionsbereich D und daher nicht alle Werte aus D auf unterschiedliche Werte in B abgebildet werden können. Die Werte in D können also ungenauer dargestellt werden, ohne daß in B ein Genauigkeitsverlust spürbar wird. Im Fall von Bereichen, in denen die erste Ableitung viel größer als 1 ($f'(x) >> 1$) ist, muß gegenüber oben gegenteilig vorgegangen werden, um die Genauigkeit im Bildbereich zu erhalten. Es müssen also die Zahlen des Definitionsbereiches genauer dargestellt werden.

Konventionelle Zahldarstellung:

S X X ... X . x x ... x

mit <------- n Bits ------>

Dabei ist S das Vorzeichen, X sind die Vorkommastellen und x die Nachkommastellen. Die kodierte Zahldarstellung ist dann:

S K ... K X ... X . x ... x

mit <---------- n Bits -------->

K ist dabei die Bereichskodierung. Für die kodierte Zahldarstellung wird eine Bereichskodierung benötigt, die feststellt, in welchem Bereich die vorliegende Zahl liegt.

Gibt es C Unterbereiche, dann beträgt die Länge der Bereichskodierung lb(C), mit lb als Zweierlogarithmus. Für die restliche Zahldarstellung bleiben dann noch n - lb(C) - 1 Stellen übrig, wenn n die Anzahl der Stellen pro Wort ist. Die 1 wird wegen dem Vorzeichen S subtrahiert.

Die Durchführung der Bereichskodierung erfolgt durch wenige Logik-Glieder (UND, ODER, NICHT), die Bildung der neuen Zahldar-

stellung geschieht durch einfaches Abschneiden und Zusammensetzen. Ausgabe aus diesem Kodierungsblock sind so viele Busse wie es Unterbereiche gibt. Die Wortbreite entspricht der Breite der externen Zahldarstellung.

Der Überlauf-Block besteht aus einer einfachen Logik, die feststellt, ob bei einer konkreten Zahl die abgeschnittenen Stellen ungleich 0 sind. Ist das der Fall, tritt ein Überlauf auf. Ausgabe des Blocks ist zunächst die in der reinen, nicht Vorzeichen behaftete Zahldarstellung größtmögliche Zahl. Ob die Zahl positiv oder negativ ist, wird durch den Zustand des Vorzeichen-Bit S angegeben.

Bei der Erfindung:

- liegt das ursprüngliche Format in verschiedenen, aber fest definierten, Festkomma-Formaten vor;
- ist das kodierte Format von den Nichtlinearitäten der im Anschluß an die Kodierung folgenden nichtlinearen Funktion abhängig und für diese Funktion optimiert;
- wird der Definitionsbereich der sich anschließenden nichtlinearen Funktion in nötigenfalls unterschiedlich große Bereiche unterteilt, die durch unterschiedliche Kodierungen voneinander unterschieden werden;
- läßt sich das elektronische System durch einen kundenspezifischen Schaltkreis (ASIC) oder durch

einen spezifischen Chip-Satz auf einer Elektronikkarte verwirklichen.

Mit dem Verfahren und dem elektronischen System zur Durchführung desselben erhält man folgende Vorteile:

- Die Genauigkeit der sich an die Kodierung anschließenden Funktion richtet sich ausschließlich nach der Breite des verwendeten Datenformats und nicht nach Nichtlinearitäten der Funktion.
- Das vorgestellte System ist sehr schnell - beim aktuellen Stand der Technik: Berechnung innerhalb eines Taktzyklus bei 50 MHz - und mit geringem Hardware-Aufwand zu realisieren, da die Durch-

führung der Kodierung keine Rechenoperationen wie Addition oder Multiplikation verlangt, sondern aus einfachen Logik-Gliedern und Multiplexern aufgebaut ist.

Die Vorteile der Genauigkeitssteigerung treten bei rechenintensiven Algorithmen zutage. Insbesondere Anwendungen aus dem Bereich der Bild- und Signalerkennung, wie Diagnosesysteme in der Medizin- oder Mikrosystemtechnik, profitieren davon.

Beispiele sind die Erkennung von Mikroverkalkungen in der weiblichen Brust bei Vorsorgeuntersuchungen (siehe W. Eppler, T. Fischer, H. Gemmeke, R. Stotzka, T. Köder, „Neural Chip SAND/1 for Real Time Pattern Recognition", IEEE Transactions on Nuclear Sciences, Vol.45, No.4, Aug 1998, pp. 1819-1823)
oder die Detektion kosmischer Teilchen (siehe W. Eppler, T. Fischer, H. Gemmeke, A. Chilingarian, A.Vardanyan, "Neural Chip SAND In Online Data Processing of Extensive Air Showers", Proceedings of 1st Int. Conf. on Modern Trends in Computational Physics, Dubna, Russia, June 1998). In beiden Fällen reicht die Rechenleistung herkömmlicher Computer nicht mehr aus. Der rechenintensive Algorithmus läuft auf einer Einsteckkarte eines PCs ab, die mit einer Festkommaarithmetik arbeitet. Gleichzeitig müssen die Ergebnisse der Berechnung sehr genau sein. Das digital-elektronische Verfahren und die Hardware-Architektur zur Durchführung desselben sind dazu sehr gut geeignet.

Die Erfindung wird im folgenden anhand der einzigen Figur der Zeichnung noch näher erläutert. Die Figur zeigt das Blockschaltbild der Formatumwandlung.

Eine Zahl x der Breite n ist im Eingabeformat $EF_f = S\ V_f\ N_f$ (Vorzeichen, Vorkommastellen, Nachkommastellen) repräsentiert. Die Vorzeichen behafteten Zahlen haben alle das Vorzeichen an der höchstwertigen Stelle. f ist ein binär kodiertes Steuerwort der Länge F, welches die Nummer des aktuellen Datenformats angibt.

Es lassen sich 2^f Eingabedaten-Formate definieren. Sie unterscheiden sich nur in der Position des Festkommas. Die Festlegung, welches Steuerwort welcher Festkomma-Position entspricht, ist frei wählbar. In einem ersten Verarbeitungsschritt werden die Bits (Al) mit den niedrigsten Werten, die später auf keinen Fall verwendet werden können, abgeschnitten. Die Bits (Ül) mit den höchsten Werten, die voraussichtlich niemals gesetzt werden, können ebenfalls abgeschnitten werden. Allerdings muß hier zur Sicherheit immer geprüft und gegebenenfalls ein Überlauf erzeugt werden.

Das Eingabeformat läßt sich dann auf eine weitere Art ausdrücken:

$$EF_f = S \quad Ül_f \quad M_f \quad Al_f.$$

Die Breite der M_f ist einheitlich m.

Die Position des Festkommas ist abhängig vom gewählten Eingabeformat EF_f. Das Zwischenformat ZF besitzt das Festkomma an einer bestimmten Stelle, unabhängig vom Eingabeformat EF_f. Dafür müssen verschiedene Bereiche M_f von EF_f an die richtige Bitposition im Zwischenformat ZF kopiert werden. Dies geschieht durch die Abbildung

$$M1: M_f \rightarrow M,$$

die in einem technischen System durch einen Multiplexer oder eine vergleichbare Logik-Schaltung erreicht wird. Der Multiplexer M1 besitzt f Eingänge zu je m Stellen, nämlich die Stellen $\{n-1-ül_f-m, \ldots, n-2-ül_f\}$. Sie werden durch ihn auf das Zwischenformat

$$ZF = S \; M$$

abgebildet.

Der Zahlenbereich, der durch das ZF-Format dargestellt wird, läßt sich in C Intervalle I_c unterteilen, so daß die Intervalle den gesamten Zahlenbereich von ZF abdecken. Überlappungen und Lücken sind nicht erlaubt.

Zum einfachen Aufbau werden als Intervallgrenzen Zweierpotenzen verwendet. Damit wird mit einfachen Logik-Gliedern für jede Zahl festgestellt, in welchem Intervall sie sich befindet. Für jede im ZF-Format repräsentierte Zahl x gilt dann:

$$x \varepsilon I_c$$

für genau einen Index C.

Das Zwischenformat ZF = S M mit der Breite m+1 ist größer als das Ausgabeformat. Deshalb entfallen wieder einige höherwertige Bits Ü2, die auf einen Überlauf hin geprüft werden müssen, und einige niederwertige Bits A2, die einfach abgeschnitten werden. Für jedes Intervall I_c werden die Schnitte an unterschiedlichen Positionen vorgenommen. Das Zwischenformat wird deshalb auch folgendermaßen definiert:

$$ZF = S \quad Ü2_c \quad B_c \quad A2_c.$$

Der Überlauf-Block stellt für die Stellen n-2 bis $n-1-ü1_f$ des Eingabeformats EF_f und für die Stellen m-1 bis $m-ü2_c$ des Zwischenformats fest, ob eine Stelle ungleich 0 ist. Ist das der Fall, wird der Überlauf Ü gesetzt. Für alle Eingabeformate EF_f und Zwischenformate ZF_c wird im Überlauf-Block folgende Operation ausgeführt:

Überlauf-Flag = 1, wenn x(i)=1 für irgendein i aus $\{n-1-ü1_f, \ldots, n-2, m-ü2_c, \ldots, m-1\}$

Überlauf-Flag = 0, sonst.

Pro Format EF_f und Zwischenformat ZF_c ist für die Operation ein logisches ODER-Glied mit $Ü1_f + Ü2_c$ Eingängen erforderlich.

Das Ausgabeformat AF als Kodierung für eine Zahl x setzt sich folgendermaßen zusammen:

$$AF = S \quad K_c \quad G_c,$$

wobei S das Vorzeichen, K_c der Bereich der Kodierung und G_c ein Ausschnitt niederwertiger Bits ist, der sich teilweise mit K_c überlappen kann. G_c ist genau so breit, daß alle Zahlen des Intervalls I_c mit der gewünschten Genauigkeit gebildet werden können. Die Intervallgröße, d.h. die Anzahl der Elemente des Intervalls, ist eine Zweierpotenz 2^d, so daß G_c die Breite d besitzt.

Die Breite von K_c ergibt sich aus der Breite des Ausgabeformats AF, abzüglich des Vorzeichenbits und der Breite von G_c. Die Kodierung K_c ist zunächst beliebig, richtet sich aber nach den Kodierungen der übrigen Intervalle, die sich gegenseitig ausschließen müssen. Alle Kodierungen K zusammen werden minimal angelegt, d.h. es gibt keine Kodierung, die keinem Intervall entspricht bzw. keine zwei Kodierungen, die das selbe Intervall repräsentieren. In beiden Fällen wird das Zahlenformat schlecht ausgenutzt und die maximal erzielbare Genauigkeit reduziert.

Die Kodierung K_c wird im Kodierungsblock aus B_c hergestellt. Dazu genügt bei günstiger Aufteilung der Intervalle die Abfrage weniger Bits von B_c, die über einfache Logik-Glieder verknüpft die Bits der neuen Kodierung K_c ergeben. Anschließend wird an K_c das Vorzeichen S und die Bits G_c mit niedrigen Werten angehängt. Dies wird im allgemeinen für jedes Intervall I_c getrennt vorgenommen, da die einzelnen Stücke unterschiedlich groß sein können. Eine Abbildung

$$M2: K_c\ G_c \rightarrow KG$$

bildet diese einzelnen Stücke auf das einheitliche Format

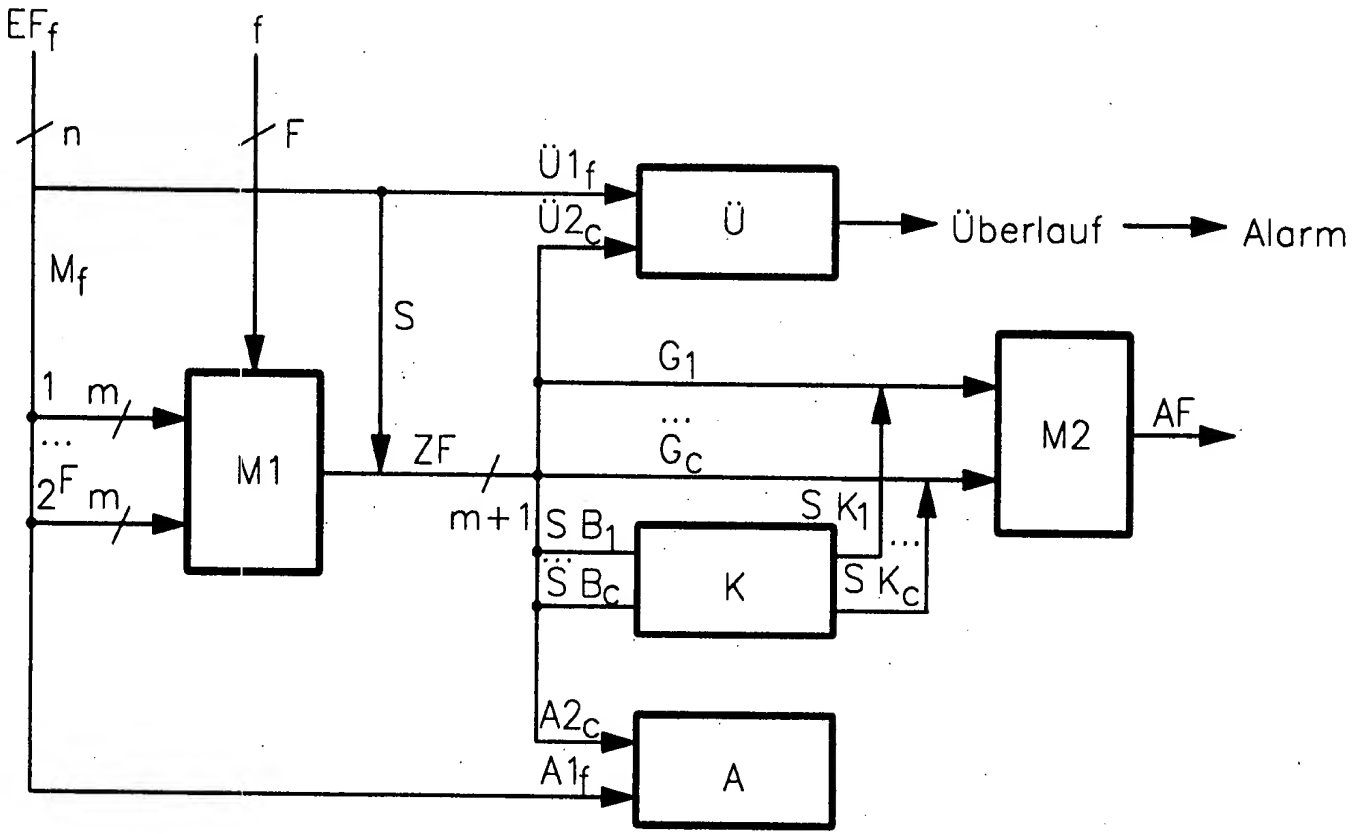
$$AF = S\ \ KG$$

ab. Auch für diese Abbildung wird hier ein Multiplexer, nämlich M2, verwendet.

Wird anschließend auf die so kodierten Zahlen eine nichtlineare Funktion angewandt, muß darauf geachtet werden, daß die Funktionsberechnung ebenfalls unter Beachtung der Intervall-Einteilung I_c erfolgt. Für jedes I_c gilt eine andere Genauigkeit, d.h. der Abstand aufeinanderfolgender Werte aus dem Definitionsbereich ist innerhalb eines Intervalls gleich, zwischen den Intervallen aber im allgemeinen ungleich. Dies ist insbesondere auch bei der Verwendung von Lookup-Tabellen zu beachten.

Zusammenfassung

Es wird ein digital-elektronisches Verfahren zur Steigerung der Berechnungsgenauigkeit bei nichtlinearen Funktionen beschrieben. Für das Verfahren ist eine Hardware-Architektur notwendig, die in ihrem Aufbau einfach, weil sehr eng an das Verfahren angepaßt, ist, wodurch dasselbe sehr preisgünstig realisierbar ist und, die spezielle Aufgabe sicher und schnell erfüllend, wirkungsvoll an die Stelle eines breit einsetzbaren jedoch vergleichsweise teueren elektronischen Systems, wie ein PC, gesetzt kann.


EF_f
f
n
F
Ü1_f
Ü2_c
Ü
Überlauf
Alarm
M_f
S
G_1
1
m
...
2^F
m
M1
ZF
M2
AF
G_c
m+1
S B_1
S K_1
...
S B_c
K
S K_c
A2_c
A1_f
A